PLOINK

GÉRARD PARESYS

JEAN-FRANÇOIS DEGREMONT

ARTINFO/MUSINFO n° 26

1 - LES ADVERSAIRES EN PRESENCE

1.1 - Le Musicien-Programmeur

1.2 - L'Instrument de Musique Formé de :

1.2.1 - LA TELETYPE

C'est la liaison Musicien-Instrument. Elle code les lettres frappées sur son clavier en trains d'impulsions électriques compréhensibles par l'INTEL.

1.2.2 - LES 4 CLES DE 8 BITS (made by D. Roncin)

C'est aussi une liaison Musicien-Instrument. Elles permettent d'intervenir en temps réel pendant l'exécution d'un programme en injectant des valeurs variables dans les registres de l'INTEL.

1.2.3 - L'INTELLEC

C'est le microprocesseur, la bête, la chose, le truc, le monstre. Pour plus de renseignements quant à sa structure et aux éléments de sa programmation voir :

POLY D'INTELGREU . Patrick GREUSSAY, Avril 1977

1.2.4 - LE 4-DAC ET LE 8-DAC A GAIN VARIABLE

L'INTEL, comme tout bon ordinateur qu'il est ne communique avec le monde qui l'entoure que par des trains d'impulsions électriques, des zéros et des uns, du courant ou pas de courant.

On dit de ce genre de fonctionnement qu'il est digital. Pourquoi pas?

Le synthétiseur VCS3 (voir plus loin), lui par contre ne connaît que des VARIATIONS de voltage, d'intensité de courant. Ce n'est plus OUI-NON mais UN PEU PLUS-UN PEU MOINS. Ce mode de fonctionnement est dit analogique.

On voit donc la nécessité d'un traducteur de langage digital en langage analogique. Tel est le rôle du 4-DAC (en clair : Convertisseur Digital Analogique à 4 voies), ou du 8-DAC (pareil, mais à 8 voies et gain variable).

1.2.5 - UN SYNTHETISEUR EMS VCS3

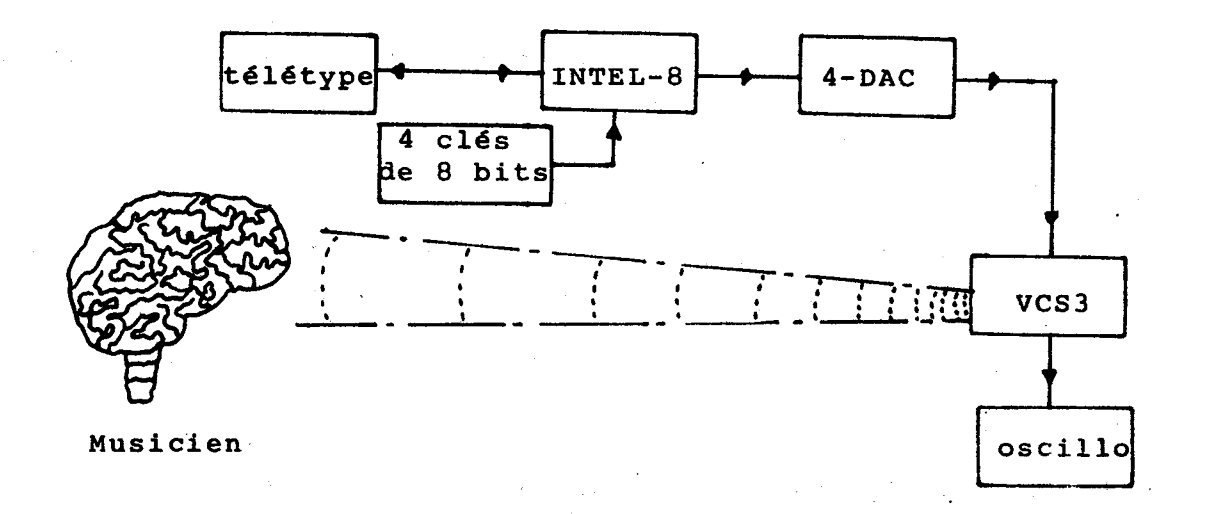
Il transforme le signal envoyé par l'INTEL en une vibration sonore. Ce miracle de la science se fait grâce à 3 oscillateurs de gammes différentes, un générateur de bruit blanc et coloré, un filtre multifonction, un modulateur à anneaux, un générateur d'enveloppes, une réverbération.

Ne pas s'affoler devant cet énoncé. Aller au labo d'informatique et toucher les boutons!! Cela viendra très vite.

Dans l'état actuel de nos connaissances et possibilités matérielles, le synthétiseur n'est pas entièrement "piloté" par l'INTEL. Toute recherche ou apport dans ce sens sont vivement appréciées.

1.2.6 - UN OSCILLOSCOPE

Il permet la visualisation sur un écran cathodique de ce qui est entendu.



2-PRELIMINAIRES

2.1 - Ce qui suit suppose que le lecteur connaît les instructions INTELGREU. Notre but est de considérer quelques programmes musicaux simples. Pour tout renseignement d'ordre théorique, consulter le poly (à paraître dans un prochain numéro d'ARTINFO/MUSINFO), son auteur, les piliers du laboratoire.

2.2 - Il est vivement recommandé après avoir lu et fait ces exercices de "passer à l'acte", de tester et de proposer d'autres exercices, de faire des concerts.

3- PREMIER PROBLEME

Définir et exécuter une mélodie. La plus simple : PIN-PON-PIN-...

3.1 - Petits détails : en utilisant le 4-DAC, à chaque note est associée une valeur constante de 00 à FF en héxa. Le 4-DAC est étalonné de la façon suivante :

8 intervalles hexa ≡ 1/2 ton

exemple :

00 ≡ DO
08 ≡ DO#
.
.
60 ≡ DO^8

Ce n'est qu'un exemple. On peut, en réglant manuellement la fréquence de sortie du VCS3 transposer cette échelle, les intervalles étant conservés.

3.2 - Autres petits détails : les 4 voies du 4-DAC s'appellent OC, OD, OE, OF. C'est une de ces voies qui sera appelée pour sortir une note de l'INTEL.

3.3 - Le problème posé en 3 devient donc : comment introduire dans la bête les valeurs associées aux notes désirées, comment les faire sortir de la bête, comment faire durer une note.

3.4 - PROGRAMME

adresse d'implantation (par exemple)	instructions		commentaires
1000	LAI 00	→	par ex. DO
1002	OUT 0C	→	par ex. voie 0C
1003	LAI 18	→	par ex. RE#
1005	OUT 0C	→	sur la même voie
1006	JMP= 1000	→	et on recommence

3.5 - Si l'INTEL-8 exécutait ce programme, cela se passerait en environ 100 microsecondes par boucle. C'est beaucoup trop rapide (mélodie répétée 10 000 fois/seconde) pour être audible. Il faut donc trouver une astuce pour ralentir tout cela.

3.5.1 - ASTUCE

On demande la sortie d'une note.

On perd du temps pendant que la note dure.

On passe à une autre note,

etc...

3.5.2 - SOUS PROGRAMME QUI PERD DU TEMPS

1100	:T	définition d'étiquette
1100	LBI FF	pour que ce soit long
1102	LCI FF	très long
1104	DCC	
1105	JFZ=1104	1ère boucle
1108	DCB	
1109	JFZ=1102	2ème boucle
110C	RET	fin de sous programme

On a fait compter l'ordinateur en décrémentant deux registres. On a imbriqué 2 boucles car 1 seule était insuffisante. Comment modifier la durée d'une note? On le fait compter moins longtemps si on veut une note plus brève, et plus longtemps si on veut une note plus longue. Donc on fait varier les valeurs hexa rentrées dans B et C.

Base : si on met FF dans B et C, chaque note temporisée durera 2,5 secondes environ.

3.6 - REPRENONS : PIN-PON TEMPORISÉ

```
1000      LAI 00
          OUT 0C
          CAL :T     <-- appel du sous programme implanté
          LAI 18                                par ailleurs
          OUT 0C
          CAL :T     <--
          JMP 1000
```

et ça marche!!

4- DEUXIEME PROBLEME

Faire exécuter par le système une octave descendante par 1/2 tons à partir de DO^4, jouée en noire à 120 et ce, répété à l'infini. Rappel pour les non musiciens : noire à 120 = 120 noires dans la minute.

Solution : on implante le programme de temporisation en 1100 (par exemple) avec les valeurs de B et C suivantes :

si FF fois FF font durer une note 2,5 secondes

$\frac{FF}{5}$ fois FF font durer une note $\frac{2,5}{5}$ secondes

$\frac{FF}{5} \simeq 33$ et $\frac{2,5}{5} = 0,5$ secondes

Or la noire à 120, c'est une note par 0,5 secondes, donc : mettre 33 dans B et FF dans C.

```
    :1
1   LEI 0D      compteur E à 0D (on va jouer 13 notes)
    LAI 60      en réglant le VCS3 pour que  60 ≡ DO4
    :2
2   OUT 0C
    CAL :T
    SUI 08      A = A - 8 (on descend de 1/2 ton)
    DCE         E = E - 1
    JTZ :1
    JMP :2
```

Et toc!

5- TROISIEME PROBLEME

Faire un glissando montant d'une 9ème majeure répété à l'infini.

Solution : - Dans le problème précédent, les intervalles étaient de 1/2 ton (rappel : sur 4-DAC 1/2 ton ≡ 08 hexa). Pour faire un glissando, on va emettre une suite de notes aussi faiblement espacées que possible : 1/16ème de ton ≡ 01 hexa

- Une 9ème majeure ≡ 70 hexa sur le 4-DAC

- Tempo : pour que le glissando soit assez rapide, on peut faire deux choses :

. soit modifier les valeurs contenues dans B et C lors de l'exécution de T (par exemple B = 02 et C = FF)

. soit commander la tempo par les 4 clés de 8 bits. Prenons par exemple la clé 04. Il suffit de modifier l'instruction 1100 de Tempo comme suit :

```
1100   :T        →   1100   :T
1100   LBI FF        1100   INI 04  → met dans A ce qui est affiché
                     1101   LBA       sur les 8 bits de la clé 04
                                    → met A dans B
```

- Programme :

```
    :1
1   LAI 20   →  pourquoi ne pas partir de 20?
    :2
2   OUT 0C
    CAL :T
    ADI 01   →  A = A + 1
    CPI 90   →  flag positionné par A-90 (90 : 20+70 : note de
                                    départ + intervalle)
    JTZ :1   →  test sur flag
    JMP :2
```

1ère remarque : on n'utilise plus un compteur de notes mais on teste si on est arrivé sur la dernière note de la séquence.

2ème remarque : certains d'entre vous ne manqueront pas de remarquer que ce programme accouplé au sous-programme Tempo modifié pour l'utilisation des clés d'accès direct ne TOURNE PAS !!

6-QUATRIEME PROBLEME

Pourquoi?

Solution : l'instruction 1100 modifiée dans Tempo écrase la valeur courante du glissando qui est dans A, donc il faut sauvegarder A dans le programme glissando avant d'appeler Tempo.

```
     :1
1    LAI 20
     :2
2    LEA      →   on sauve A dans E
     OUT 0C
     CAL :T   →   A est écrasé pendant T
     LAE      →   on récupère A dans E
     ADI 01
     CPI 90
     JTZ :1
     JMP :2
```

7-CINQUIEME PROBLEME

Construire un programme qui génère des mélodies aléatoires.

2 paramètres : - le rythme
- la hauteur des notes

Solution : dans le programme il y a un sous-programme qui crée un nombre aléatoire qui servira à définir soit une hauteur, soit une durée de note.

Le voilà :

```
:A
ADA
RFS
XRI 7D
RET
```

merci Monsieur GREUSSAY.

Servons nous en

```
LAI 08    → par exemple pour initialiser l'aléatoire
:1
CAL :A
OUT 0C
CAL :T
JMP :1
```

C'est facile : cela sort des notes de hauteur aléatoire et de même durée.

8-SIXIEME PROBLEME

Compliquons : varions la durée.
Il faut modifier le Tempo (cf. 3.5.2) pour que la longueur d'une note puisse être "imposée" de l'extérieur du sous-programme.

Solution :

```
1100 :T
1100 LCI FF
1102 DCC
1103 JFZ=1102
1106 DCB
1107 JFZ=1100
110A RET
```

et il faudra charger B de l'extérieur, par exemple avec une valeur aléatoire.

PROGRAMME ALEATOIRE EN HAUTEUR ET EN DUREE

```
        LAI   08
        :1
        CAL   :A
        OUT   0C
(*)     LBA
        CAL   :T
        JMP   :1
```

On voit que le même aléatoire est utilisé pour hauteur et durée. On a donc une corrélation dans la mélodie : les aigües sont plus longues et les graves sont plus courtes.

On peut vouloir la corrélation inverse, pour ce faire on temporise par (FF - valeur de A).

On remplace à partir (*) par :

```
LEA
LAI   FF
SUE

LBA
CAL   :T
JMP   :1
```

c'est bestial, mais ça marche!!

9 - SEPTIEME PROBLEME

On peut aussi vouloir rythme et durée complètement indépendants.

Solution

On rappelle :A avant de temporiser.
A partir de (*) remplacer par :

```
CAL  :A
LBA
CAL  :T
JMP  :1
```

On peut aussi vouloir des aléatoires, mais seulement certaines aléatoires. Si on prend n'importe quelle aléatoire, les hauteurs correspondantes peuvent prendre toutes les valeurs possibles de 16° de ton en 16° de ton, sur toute l'échelle du DAC.

Faisons un filtre qui ne laissera sortir que des notes espacées, par exemple de 1/8 de ton.

Après chaque appel CAL :A pour les hauteurs ajouter l'instruction NDI FE.

Explication : FE est en binaire 1111 1110
NDI est l'instruction "ET logique bit à bit" elle met systématiquement le dernier bit à droite à 0. Cela transforme les aléatoires dont le dernier bit est à 1 en aléatoire avec 0 à droite, donc plus de 16° de ton.

De même, pour avoir des mélodies en :

1/4 de ton faire	NDI	FC	FC = 1111 1100
1/2 ton faire	NDI	F8	F8 = 1111 1000
1 ton faire	NDI	F0	F0 = 1111 0000
en tierce majeure faire	NDI	E0	E0 = 1110 0000
en quinte augmentée "	NDI	C0	C0 = 1100 0000

10 - HUITIEME PROBLEME

Maintenant le luxe des luxes : **un séquenceur**

Solution :

```
LHI  12
LLI  00
:1
LAM
OUT  0C
CAL  :T     → celui de 3.5.2
INL
JMP  :1
```

On lit toute la mémoire à partir de l'adresse 1200 jusqu'à la fin des temps (à la fin de la mémoire, il repart au début de la mémoire). Si on a placé à partir de 1200 les valeurs correspondantes à la mélodie désirée, cela l'exécute.

Supposons désirée une mélodie de n notes (n ≤ 256) répétée indéfiniment

```
:0
LHI  12
LLI  00
:1
LAM
OUT  0C
CAL  :T
INL
LAI  n       ←   le nombre de notes de la mélodie
SUL
JFZ  :1
JMP  :0
```

Exercice : faire un séquenceur où les durées des notes ne sont ni constantes, ni aléatoires (répétitif en hauteur et durée).

11 - NEUVIEME PROBLEME

Faire une mélodie répétitive en hauteur, mais pas en durée.

Solution : tempo aléatoire

```
:0
LHI  12
LLI  00
:1
LAM
OUT  0C
CAL  :A
LBE
```

```
CAL  :T       ←      tempo du 6ème problème
INL
LAI  n
SUL
JFZ  :1
JMP  :0
```

Il faut modifier l'aléatoire de telle sorte que la valeur courante ne soit pas dans A (erreur de même type que celle du 4ème problème).

```
:A
LAE
ADA
LEA
RFS
XRI  7D
LEA
RET
```

12 - DIXIEME PROBLEME

Do it yourself ...